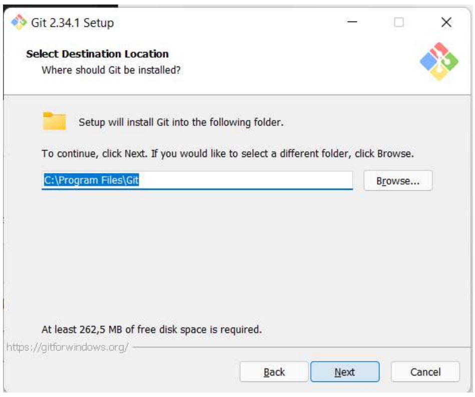
From sourcecode to a running M5Stack on Windows

Mittwoch, 29. Dezember 2021 11:15

First of all, Visual Studio Code, also named VSCode has nothing to do with Microsoft Visual Studio!

Download and install VSCode:
https://code.visualstudio.com/download

Download and install GIT:
https://git-scm.com/



Select Destination Location
Where should Git be installed?

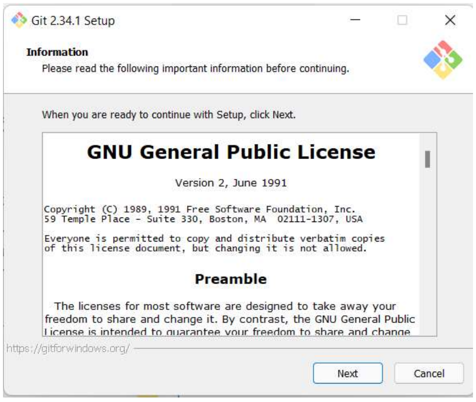
Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

At least 262,5 MB of free disk space is required.

https://gitforwindows.org/



Information
Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

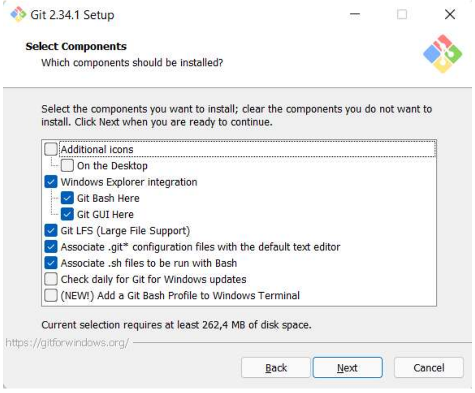
```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

https://gitforwindows.org/



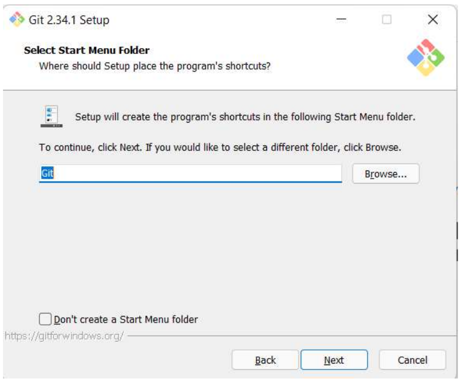
Select Components
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- □ Additional icons
  - □ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Git Bash Here
  - ☑ Git GUI Here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Check daily for Git for Windows updates
- □ (NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 262,4 MB of disk space.

https://gitforwindows.org/



Git 2.34.1 Setup

Select Start Menu Folder

Where should Setup place the program's shortcuts?

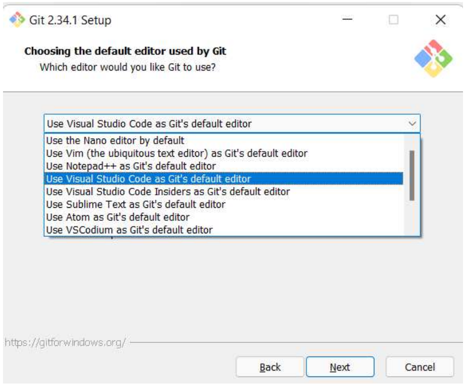
Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Git

□ Don't create a Start Menu folder

https://gitforwindows.org/



Git 2.34.1 Setup

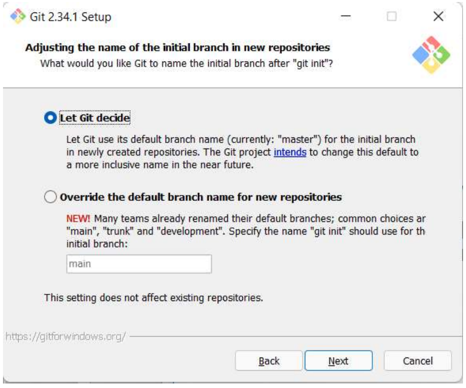
Choosing the default editor used by Git

Which editor would you like Git to use?

Use Visual Studio Code as Git's default editor

- Use the Nano editor by default
- Use Vim (the ubiquitous text editor) as Git's default editor
- Use Notepad++ as Git's default editor
- Use Visual Studio Code as Git's default editor
- Use Visual Studio Code Insiders as Git's default editor
- Use Sublime Text as Git's default editor
- Use Atom as Git's default editor
- Use VSCodium as Git's default editor

https://gitforwindows.org/



Git 2.34.1 Setup

Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"?

● **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.

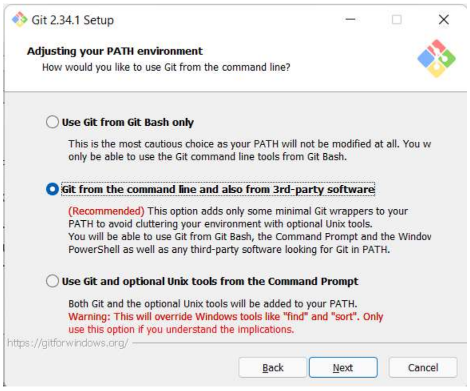
○ **Override the default branch name for new repositories**

NEW! Many teams already renamed their default branches; common choices ar "main", "trunk" and "development". Specify the name "git init" should use for th initial branch:

main

This setting does not affect existing repositories.

https://gitforwindows.org/



Git 2.34.1 Setup

Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You w only be able to use the Git command line tools from Git Bash.

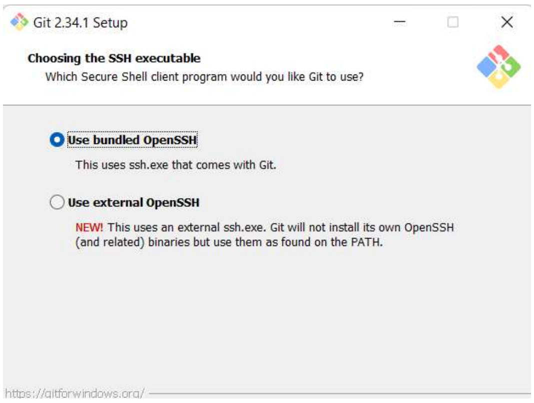
● **Git from the command line and also from 3rd-party software**

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windov PowerShell as well as any third-party software looking for Git in PATH.

○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH. Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/



Git 2.34.1 Setup

Choosing the SSH executable

Which Secure Shell client program would you like Git to use?

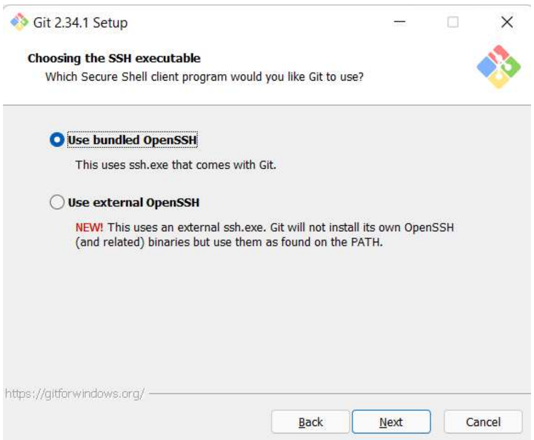
● **Use bundled OpenSSH**

This uses ssh.exe that comes with Git.

○ **Use external OpenSSH**

NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.

https://gitforwindows.org/



Choosing the SSH executable

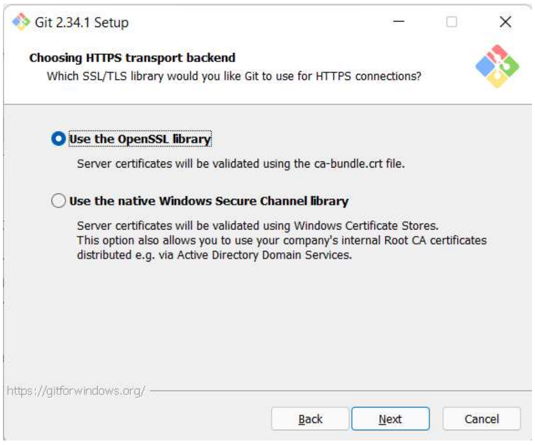
Which Secure Shell client program would you like Git to use?

● **Use bundled OpenSSH**

This uses ssh.exe that comes with Git.

○ **Use external OpenSSH**

NEW! This uses an external ssh.exe. Git will not install its own OpenSSH
(and related) binaries but use them as found on the PATH.



Choosing HTTPS transport backend

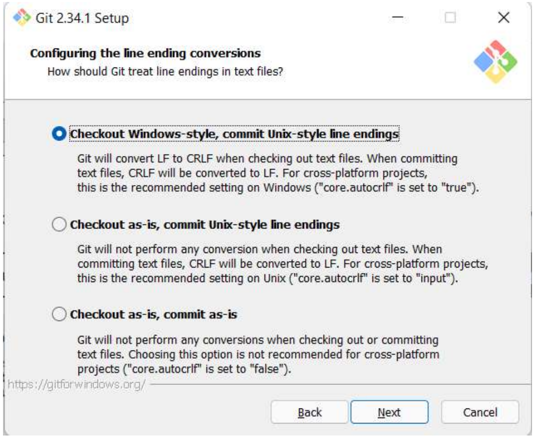
Which SSL/TLS library would you like Git to use for HTTPS connections?

● **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates
distributed e.g. via Active Directory Domain Services.



Configuring the line ending conversions

How should Git treat line endings in text files?

● **Checkout Windows-style, commit Unix-style line endings**

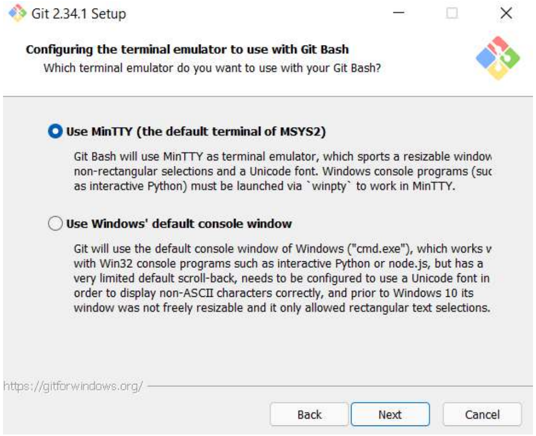
Git will convert LF to CRLF when checking out text files. When committing
text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Windows ("core.autocrlf" is set to "true").

○ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When
committing text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing
text files. Choosing this option is not recommended for cross-platform
projects ("core.autocrlf" is set to "false").



Configuring the terminal emulator to use with Git Bash

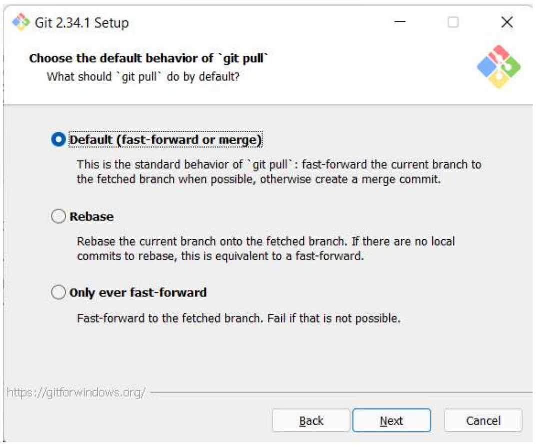
Which terminal emulator do you want to use with your Git Bash?

● **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window
non-rectangular selections and a Unicode font. Windows console programs (suc
as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works v
with Win32 console programs such as interactive Python or node.js, but has a
very limited default scroll-back, needs to be configured to use a Unicode font in
order to display non-ASCII characters correctly, and prior to Windows 10 its
window was not freely resizable and it only allowed rectangular text selections.



Choose the default behavior of `git pull`

What should `git pull` do by default?

● **Default (fast-forward or merge)**

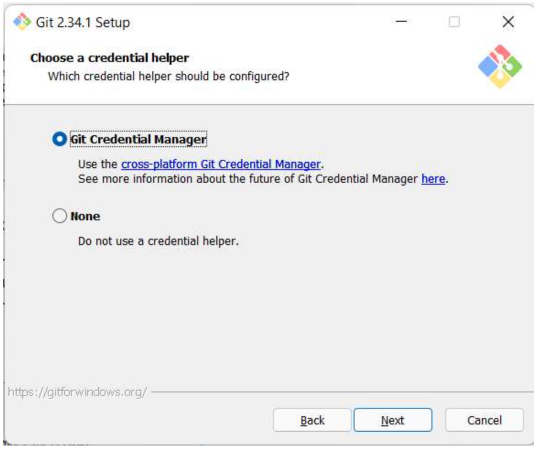
This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

○ **Rebase**

Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

○ **Only ever fast-forward**

Fast-forward to the fetched branch. Fail if that is not possible.



Choose a credential helper

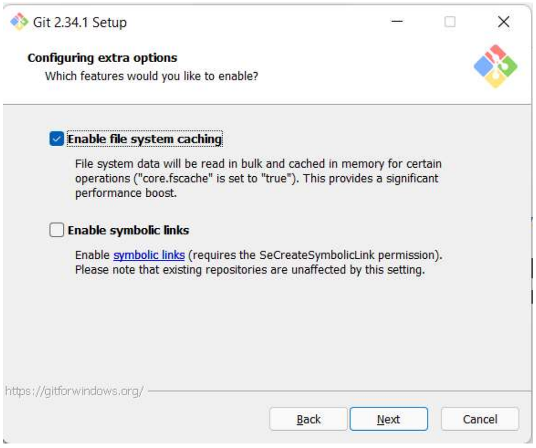
Which credential helper should be configured?

● **Git Credential Manager**

Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.

○ **None**

Do not use a credential helper.



Configuring extra options

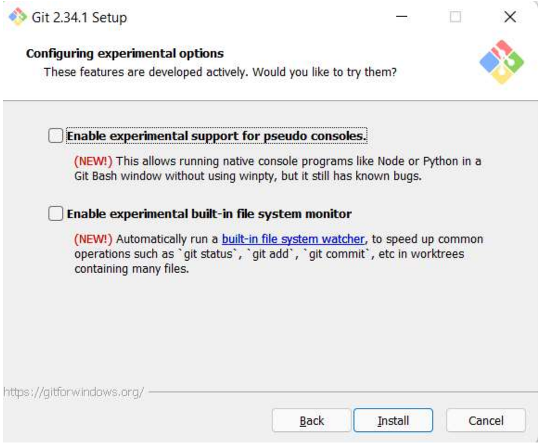
Which features would you like to enable?

☑ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☐ **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.



Configuring experimental options

These features are developed actively. Would you like to try them?

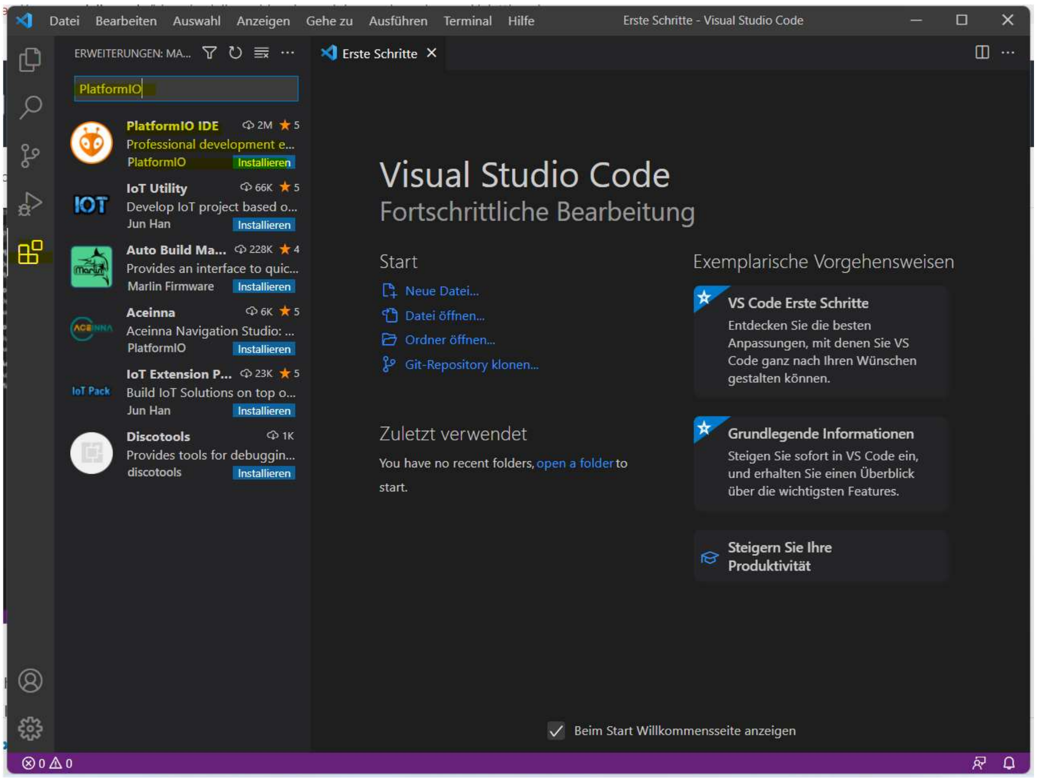
☐ **Enable experimental support for pseudo consoles.**

(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.

☐ **Enable experimental built-in file system monitor**

(NEW!) Automatically run a built-in file system watcher, to speed up common operations such as `git status`, `git add`, `git commit`, etc in worktrees containing many files.

Start VSCode and install the PlatformiO Extension:


Datei Bearbeiten Auswahl Anzeigen Gehe zu Ausführen Terminal Hilfe
Erste Schritte - Visual Studio Code
ERWEITERUNGEN: MA...
Erste Schritte
PlatformIO
PlatformIO IDE 2M ★5
Professional development e...
PlatformIO Installieren
IoT Utility 66K ★5
Develop IoT project based o...
Jun Han Installieren
Auto Build Ma... 228K ★4
Provides an interface to quic...
Marlin Firmware Installieren
Aceinna 6K ★5
Aceinna Navigation Studio: ...
PlatformIO Installieren
IoT Extension P... 23K ★5
Build IoT Solutions on top o...
Jun Han Installieren
Discotools 1K
Provides tools for debuggin...
discotools Installieren
Visual Studio Code
Fortschrittliche Bearbeitung
Start
Neue Datei...
Datei öffnen...
Ordner öffnen...
Git-Repository klonen...
Zuletzt verwendet
Exemplarische Vorgehensweisen
VS Code Erste Schritte
Grundlegende Informationen
Steigern Sie Ihre Produktivität
☑ Beim Start Willkommensseite anzeigen
⊗ 0 ⚠ 0

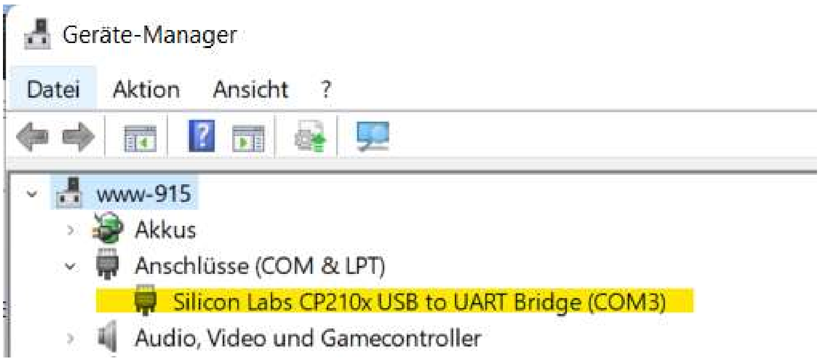
You have no recent folders, open a folder to start.

Entdecken Sie die besten Anpassungen, mit denen Sie VS Code ganz nach Ihren Wünschen gestalten können.

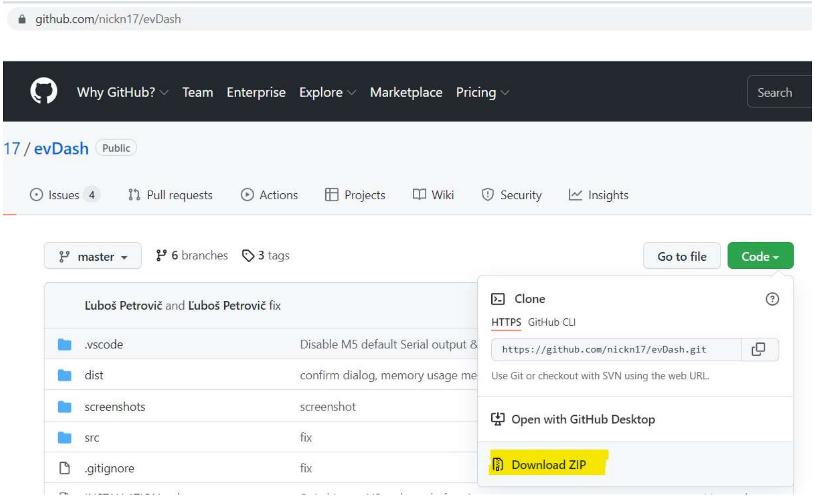
Steigen Sie sofort in VS Code ein, und erhalten Sie einen Überblick über die wichtigsten Features.

Download and install the USB to UART driver (CP2104). It can be downloaded here: https://shop.m5stack.com/pages/download

If it is installed, figure out which COM port it uses, we need this later:


Geräte-Manager
Datei Aktion Ansicht ?
www-915
Akkus
Anschlüsse (COM & LPT)
Silicon Labs CP210x USB to UART Bridge (COM3)
Audio, Video und Gamecontroller

Download the evDash source code from https://github.com/nickn17/evDash

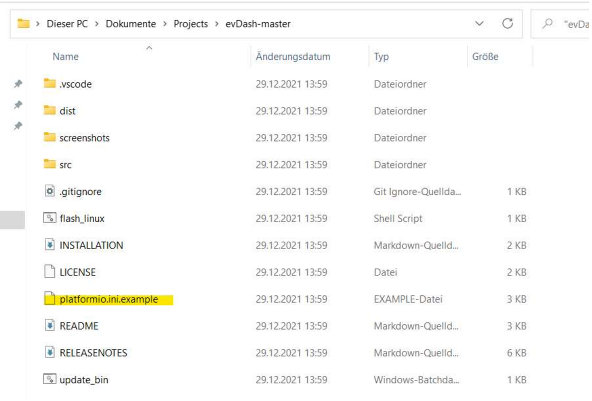

github.com/nickn17/evDash
Why GitHub? Team Enterprise Explore Marketplace Pricing
Search
17 / evDash Public
Issues 4 Pull requests Actions Projects Wiki Security Insights
master 6 branches 3 tags
Go to file Code
Ľuboš Petrovič and Ľuboš Petrovič fix
.vscode
dist
screenshots
src
.gitignore
Clone
HTTPS GitHub CLI
Open with GitHub Desktop
Download ZIP

.vscode	Disable M5 default Serial output &
dist	confirm dialog, memory usage me
screenshots	screenshot
src	fix
.gitignore	fix

```
https://github.com/nickn17/evDash.git
```

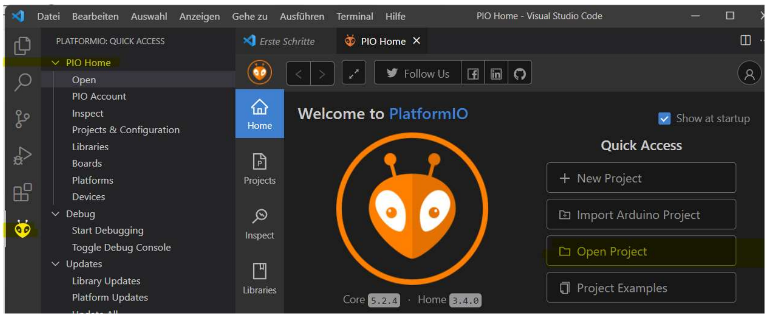
Use Git or checkout with SVN using the web URL.

Extract the ZIP Archiv and rename the file platformio.ini.example to platformio.ini

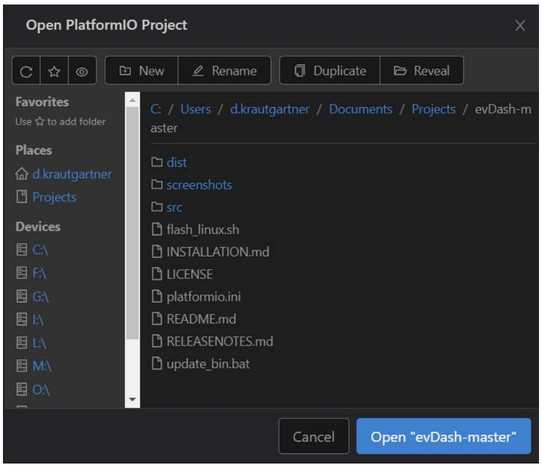

Dieser PC › Dokumente › Projects › evDash-master

Name	Änderungsdatum	Typ	Größe
.vscode	29.12.2021 13:59	Dateiordner	
dist	29.12.2021 13:59	Dateiordner	
screenshots	29.12.2021 13:59	Dateiordner	
src	29.12.2021 13:59	Dateiordner	
.gitignore	29.12.2021 13:59	Git Ignore-Quellda...	1 KB
flash_linux	29.12.2021 13:59	Shell Script	1 KB
INSTALLATION	29.12.2021 13:59	Markdown-Quelld...	2 KB
LICENSE	29.12.2021 13:59	Datei	2 KB
platformio.ini.example	29.12.2021 13:59	EXAMPLE-Datei	3 KB
README	29.12.2021 13:59	Markdown-Quelld...	3 KB
RELEASENOTES	29.12.2021 13:59	Markdown-Quelld...	6 KB
update_bin	29.12.2021 13:59	Windows-Batchda...	1 KB

Open VSCode, klick on Platformio -> PIO Home -> Open -> Open Project:



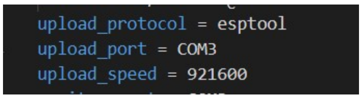
Navigate to the folder where platformio.ini is stored and click on "Open evdash-master":


Open PlatformIO Project

C: / Users / d.krautgartner / Documents / Projects / evDash-master

- dist
- screenshots
- src
- flash_linux.sh
- INSTALLATION.md
- LICENSE
- platformio.ini
- README.md
- RELEASENOTES.md
- update_bin.bat

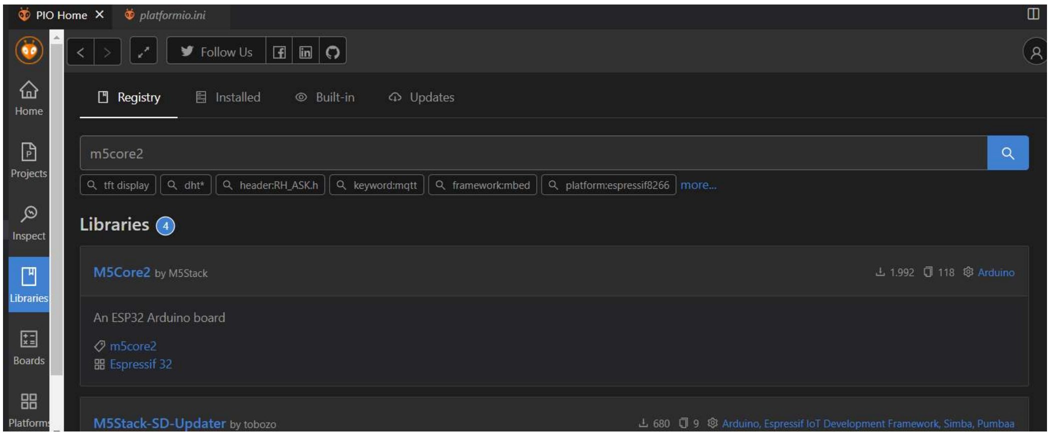
If the project is open you see the platformio.ini. You have to set the right COM Port:


```
upload_protocol = esptool
upload_port = COM3
upload_speed = 921600
```

Save the file and restart VSCode.

The first time I tried to compile the source code, it did not work because the M5Core2 library was missing.

I had to add it here:



m5core2

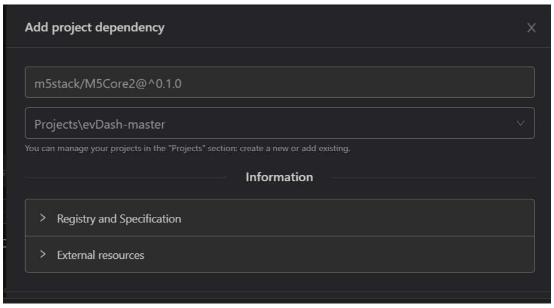
M5Core2 by M5Stack

An ESP32 Arduino board

m5core2
Espressif 32

M5Stack-SD-Updater by tobozo

Click on "M5Core2":
Click on "Add to Project:



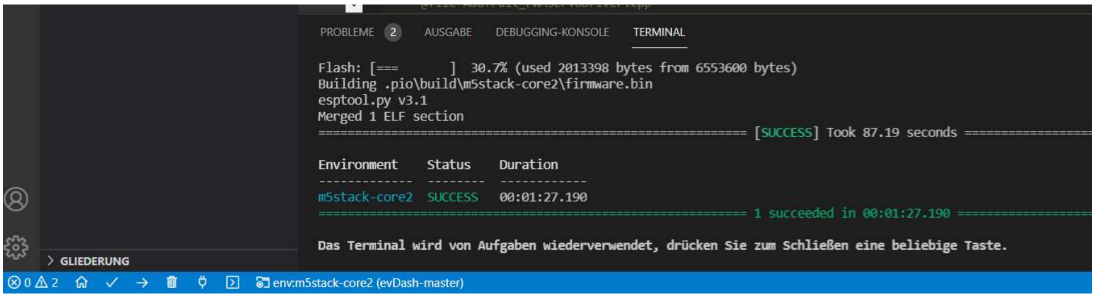
Add project dependency

m5stack/M5Core2@^0.1.0

Projects\evDash-master

You can manage your projects in the "Projects" section: create a new or add existing.

Now we are ready to go. In the blue line on the bottom we can select the right enviroment.
Click on env: In my case I selected m5stack-core2.
Then click on the checkmark. The output should look like this:

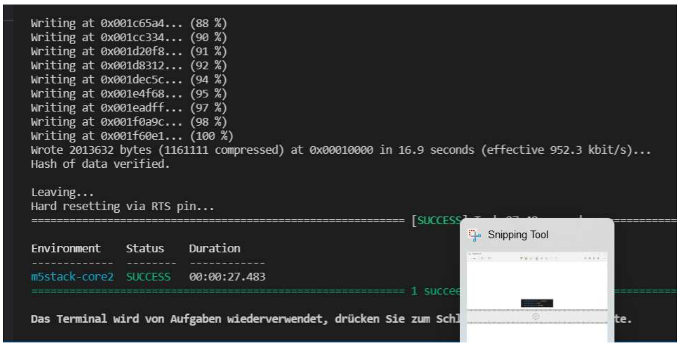


```
Flash: [===       ]  30.7% (used 2013398 bytes from 6553600 bytes)
Building .pio\build\m5stack-core2\firmware.bin
esptool.py v3.1
Merged 1 ELF section
============================================================ [SUCCESS] Took 87.19 seconds ==================

Environment    Status    Duration
-------------  --------  ------------
m5stack-core2  SUCCESS   00:01:27.190
============================================================ 1 succeeded in 00:01:27.190 ==================

Das Terminal wird von Aufgaben wiederverwendet, drücken Sie zum Schließen eine beliebige Taste.
```

When compiling was succsessful, we can flash our M5Stack by clicking on the arrow right from the checkmark in the blue line:



```
Writing at 0x001c65a4... (88 %)
Writing at 0x001cc334... (90 %)
Writing at 0x001d20f8... (91 %)
Writing at 0x001d8312... (92 %)
Writing at 0x001dec5c... (94 %)
Writing at 0x001e4f68... (95 %)
Writing at 0x001eadff... (97 %)
Writing at 0x001f0a9c... (98 %)
Writing at 0x001f60e1... (100 %)
Wrote 2013632 bytes (1161111 compressed) at 0x00010000 in 16.9 seconds (effective 952.3 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
============================================================ [SUCCESS

Environment    Status    Duration
-------------  --------  ------------
m5stack-core2  SUCCESS   00:00:27.483
============================================================ 1 succee

Das Terminal wird von Aufgaben wiederverwendet, drücken Sie zum Schl                        te.
```

Before you use the terminal I recommend to do the following settings on the M5stack:

Menu -> Adapter -> CAN
(If BLE is selected serial console does not work and device reboots every time you try to type.)

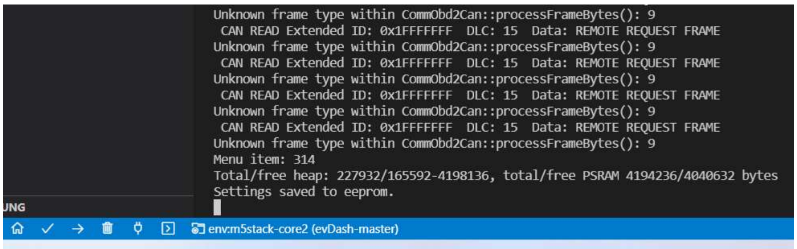
Menu -> others -> Debug level "Net"
(other debug levels create very much output...)

Now you can connect to the device via terminal. On the blue line in VSCode click on the plug sign (serial monitor) and you will see the output from the device.

Now we can do some settings via terminal:

You don't see what you type!

In some cases you see a response. For example after you typed "saveSettings" you will get "Settings saved to eeprom"



```
Unknown frame type within CommObd2Can::processFrameBytes(): 9
 CAN READ Extended ID: 0x1FFFFFFF  DLC: 15  Data: REMOTE REQUEST FRAME
Unknown frame type within CommObd2Can::processFrameBytes(): 9
 CAN READ Extended ID: 0x1FFFFFFF  DLC: 15  Data: REMOTE REQUEST FRAME
Unknown frame type within CommObd2Can::processFrameBytes(): 9
 CAN READ Extended ID: 0x1FFFFFFF  DLC: 15  Data: REMOTE REQUEST FRAME
Unknown frame type within CommObd2Can::processFrameBytes(): 9
 CAN READ Extended ID: 0x1FFFFFFF  DLC: 15  Data: REMOTE REQUEST FRAME
Unknown frame type within CommObd2Can::processFrameBytes(): 9
Menu item: 314
Total/free heap: 227932/165592-4198136, total/free PSRAM 4194236/4040632 bytes
Settings saved to eeprom.
```

wifiSsid=MyWLANSSID
wifiPassword=verysecret
abrpApiToken=abc123
saveSettings
reboot

Then goto menu -> others -> Wifi -> Wifi enabled "on"

Now you should see the IP Address in the WLAN menu.

It is always a good idea to have the terminal open for:

saveSettings
Reboot

If you want ABRP to work you have to set the following on the device:

others -> rem. Upload -> and set:
Module Type wifi
API Uload int. Off
ABRP Upload int 10-120 sec.

Don't forget to save!